

소벨 연산을 이용한 FPGA 기반 고속 윤곽선 검출 회로 구현

류상문*

FPGA-based Implementation of Fast Edge Detection using Sobel Operator

Sang-Moon Ryu*

*Professor, Department of Information and Control Engineering, Kunsan National University, Gunsan, 54150 Korea

요 약

영상에 포함된 객체의 인식을 위해서는 영상에 대한 윤곽선 검출이 선행되어야 한다. 윤곽선 검출 연산이 하드웨어로 수행되면 그 수행 시간이 소프트웨어로 구현된 경우보다 비교할 수 없을 만큼 감소하게 된다. 윤곽선 검출을 위한 연산 중 하드웨어 구현에 적합한 연산은 소벨 연산이며, 소벨 연산을 효율적으로 FPGA로 구현하기 위한 많은 연구가 수행되었다. 본 논문에서는 소벨 연산을 FPGA로 구현하기 위한 기존의 구조를 개선하여, 약간의 추가적인 하드웨어 자원의 사용만으로 그 성능을 개선할 수 있는 회로 구조를 제안한다. 제안된 구조는 윤곽선 검출 대상 영상이 메모리에 저장되어 있는 경우에 적합하며 기존의 방법 대비 약 2배의 성능 향상을 이룰 수 있다.

ABSTRACT

The edges of image should be detected first so that the objects in the image can be identified. An hardware-implemented edge detection algorithm outperforms its software version. Sobel operation is the most suitable algorithm for an hardware implementation of edge detection. And lots of works have been done to perform Sobel operations efficiently on FPGA-based hardware. This work proposes how to implement fast edge detection circuit on FPGA, which is based on the conventional circuit for edge detection using Sobel operator. The newly proposed circuit is suitable for processing images when the images are stored in memory devices and outperforms the conventional one with little additional FPGA resources. Both the conventional circuit and the proposed circuit were implemented on an FPGA. And the result showed that the proposed circuit almost doubles the performance in processing images and needs little additional FPGA resources.

키워드 : 영상 처리, 윤곽선 검출, 소벨 연산, FPGA

Keywords : Image processing, Edge detection, Sobel operation, FPGA

Received 5 July 2022, Revised 14 July 2022, Accepted 20 July 2022

* Corresponding Author Sang-Moon Ryu(E-mail:smryu@kunsan.ac.kr, Tel:+82-63-469-4688)

Professor, Department of Information and Control Engineering, Kunsan National University, Gunsan, 54150 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.8.1142>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

영상에 포함된 물체를 분할하고 분할된 물체를 인식하기 위해서는 윤곽선(edge) 검출이 필수적이다. 윤곽선 검출은 영상에 포함된 물체의 경계에서는 화소(pixel)들의 화소값(gray level)에 급격한 변화가 발생한다는 사실을 이용하며, 모든 윤곽선 검출 알고리즘은 화소값의 변화량을 계산하여 변화량이 특정 기준값을 초과하는 곳을 경계로 판단하는 방식을 택한다.

윤곽선 검출 알고리즘에서 사용되는 대표적인 연산으로 캐니(Canny) 연산[1, 2]과 소벨(Sobel) 연산[3-5]을 들 수 있다. 캐니 연산은 가장 좋은 성능을 얻을 수 있지만, 구현이 복잡하고 필요한 계산량이 많아 고속처리가 요구되는 분야에는 적용에 제한이 따른다. 이에 반해 소벨 연산은 윤곽선 검출 성능은 캐니 연산보다 떨어지지만 1차 미분 연산만을 이용하기 때문에 필요한 계산량이 현저하게 적다. 따라서 고속처리를 위한 하드웨어로 구현하기가 캐니 연산보다 매우 유리하다.

소벨 연산자를 이용한 윤곽선 검출을 고속으로 수행하기 위해 FPGA(Field Programmable Gate Array)를 이용한 하드웨어로 구현하는 방안은 대부분 초기 연구 [6]에서 제안된 회로 구조를 기반으로 하여 연구되었다. 이 회로 구조를 기반으로 하여 이전 연구 [7-15]에서는 응용 분야의 필요 또는 제약에 따라 FPGA를 이용한 효율적 회로의 구현과 이에 따른 성능 분석을 수행하였다.

본 연구에서는 이전 연구들과 달리 초기 연구 [6]에서 제안된 회로 구조를 개선하여, 추가적인 하드웨어 자원의 사용은 최소로 하면서 그 성능을 개선할 수 있는 회로 구조를 제안한다. 기존의 회로와 제안된 회로를 모두 FPGA로 구현하여 두 가지 경우에 대한 FPGA 자원 사용률과 회로의 동작 속도를 분석, 비교하여 제안된 방법의 효과를 입증한다.

II. 소벨 연산 및 하드웨어 구현

소벨 연산을 이용한 윤곽선 검출은 그림 1의 소벨 마스크(mask)[3-5]를 서로 인접한 9개의 화소들에 적용하여 이들에 대한 가로 방향의 화소값 변화량과 세로 방향의 화소값 변화량을 계산하여 이루어진다. 그림 2와 같이 수평 해상도가 M , 수직 해상도가 N 인 영상의 최좌

측상단에 위치한 9개의 화소에 대해 소벨 연산을 적용하면 가로 방향의 화소값 변화량 G_x 와 세로 방향의 화소값 변화량 G_y 는 각각 그림 1의 (a)와 (b)를 적용하여 식 (1)과 (2)처럼 계산하고, 이들로부터 총 화소값 변화량 G 를 식 (3) 또는 식 (4)을 이용해 계산한다. 그림 2와 식 (1)과 (2)에서 P_{nm} 은 각 화소의 화소값을 의미하며, 구해진 총 화소값 변화량이 설정된 기준값을 초과하는 지에 따라서 마스크가 적용된 9개의 화소들 중 중심에 위치한 화소가 경계를 의미하는지 여부를 결정한다.

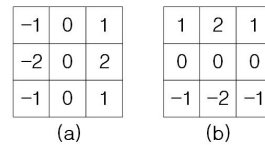


Fig. 1 Sobel operation mask

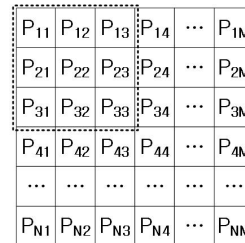


Fig. 2 Application of Sobel operation on an image

$$G_x = (P_{11} - P_{13}) + 2(P_{21} - P_{23}) + (P_{31} - P_{33}) \quad (1)$$

$$G_y = (P_{11} - P_{31}) + 2(P_{12} - P_{32}) + (P_{13} - P_{33}) \quad (2)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$G = |G_x| + |G_y| \quad (4)$$

위의 과정을 그림 3과 같이 입력 영상의 모든 화소들에 대해 적용하면 윤곽선만을 포함하는 출력 영상을 얻게 되며, 출력 영상의 수평, 수직 해상도는 입력 영상의

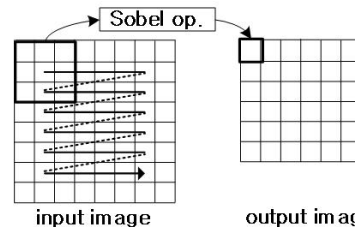


Fig. 3 Conceptual process of edge detection

그것보다 2씩 작아진다.

소벨 연산을 이용한 윤곽선 검출을 하드웨어로 구현하는 기본적인 방법은 이전 연구 [6]에서 소개된 그림 4과 같은 구조를 이용하는 것이다. 그림 4의 구조에는 9개의 레지스터(정사각형)와 두 개의 FIFO(First-in, First-out) 그리고 소벨 연산 회로가 포함되어 있다. 입력 영상의 수평 해상도를 M이라고 하면 각 FIFO의 크기는 (M-3)이다. 화소들의 화소값은 $P_{11}, P_{12}, P_{13}, P_{14}, \dots$ 의 순서로 순차적으로 입력되어 레지스터와 FIFO를 거쳐서 일시적으로 저장되며, 소벨 연산이 적용될 9개의 화소값들이 레지스터에 저장되었을 때마다 소벨 연산이 수행된다. 그림 3처럼 입력 영상 위에 소벨 마스크를 이동하면서 소벨 연산을 수행하는 대신, 입력 영상의 화소값들이 순차적으로 그림 4의 레지스터들과 FIFO에 저장되면서 소벨 연산이 적용되는 것이다. 그림 4는 그림 2에서 점선 사각형으로 표시된 화소들에 대한 소벨 연산이 수행되는 순간을 보여준다.

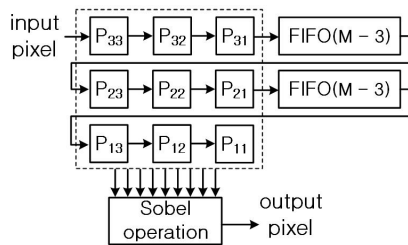


Fig. 4 Hardware implementation of edge detection using Sobel operation

최근에는 대부분의 FPGA에는 상당량의 SRAM이 포함되어 있기 때문에 이와 같은 구조의 회로를 구현하는 것이 어렵지 않으며, 회로 구현에 파이프라인(pipelining) [16]을 적용하면 평균적으로 매 클럭마다 1회의 소벨 연산을 수행할 수 있다.

III. 제안된 FPGA 기반 구현 방법

3.1. 회로 구조 및 동작

그림 5는 소벨 연산을 이용한 FPGA 기반 고속 윤곽선 검출 회로의 구성도를 보여준다. 제안된 회로는 두 개의 입력과 두 개의 출력, 16개의 레지스터와 크기가 (M/2-2)인 4개의 FIFO 그리고 두 개의 소벨 연산 회로를

포함한다. 제안된 회로를 이용하여 그림 2의 화소들에 대해 소벨 연산을 수행하는 과정은 다음과 같다.

그림 6처럼 입력 1에는 $P_{11}, P_{13}, P_{15}, P_{17}, \dots$ 의 순서로 화소값이 공급되고, 이와 동시에 입력 2에는 $P_{12}, P_{14}, P_{16}, P_{18}, \dots$ 의 순서로 화소값이 공급된다. 순차적으로 입력1과 입력2에 화소값들이 공급되어 화소값 P_{11} 이 마지막 레지스터에 저장되면 그림 6와 같이 모든 레지스터들과 FIFO들이 화소값으로 채워지게 된다. 이때 그림 7에 묘사된 것처럼 화소값 ($P_{11}, P_{12}, P_{13}, P_{21}, P_{22}, P_{23}, P_{31}, P_{32}, P_{33}$)과 ($P_{12}, P_{13}, P_{14}, P_{22}, P_{23}, P_{24}, P_{32}, P_{33}, P_{34}$)에 대한 각각의 소벨 연산을 동시에 수행하여 총 화소값 변화량 G_{11} 과 G_{12} 을 구하여, 출력 영상의 화소값 Q_{11} 과 Q_{12} 을 구한다. 이 과정은 입력 영상의 모든 화소값들이 입력 1과 2에 순차적으로 공급되면서 반복적으로 수행된다.

제안된 방식은 윤곽선 검출 대상 영상이 이미 메모리에 저장되어 있고 이를 하드웨어를 이용하여 처리하려는 상황에 적합하다. 메모리에 저장되어 있는 영상의 인접한 두 개 화소의 화소값을 동시에 읽어서 제안된 구조의 회로에 병렬 공급하는 것은 쉽게 구현될 수 있다. 회로 구현에 파이프라인을 적용하면 평균적으로 매 클럭마다 2회의 소벨 연산을 수행할 수 있기 때문에 그림 4의 회로에 비해 대략 2배 정도의 빠른 성능을 구현할 수 있다. 기존 회로 대비 제안된 회로에 추가되는 주요 구성 요소는 1개의 소벨 연산 회로, 3개의 레지스터, 입출력 인터페이스 회로, 2개의 FIFO를 4개로 분리하기 위한 주변 회로 정도이다. FIFO의 총 저장 용량은 $2(M-3)$ 에서 $4(M/2-2)$ 로 매우 근소하게 감소된다.

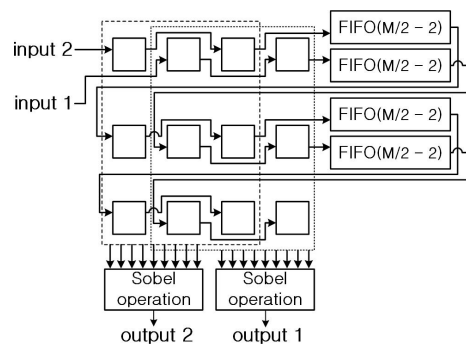


Fig. 5 Block diagram of the proposed design

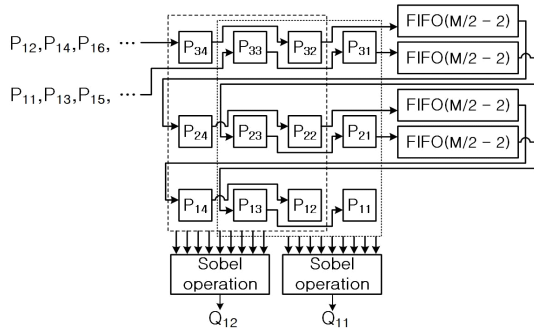


Fig. 6 Operation of the proposed design

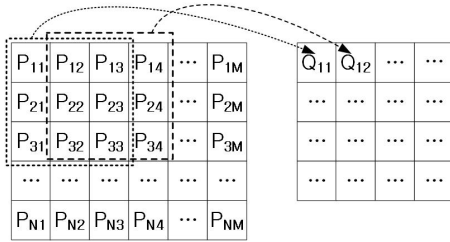


Fig. 7 Sobel operations of the proposed design

3.2. 구현 및 분석

기존 방식인 그림 4의 회로와 제안된 방식인 그림 5의 회로를 Xilinx사의 FPGA 개발 도구인 Vivado HLS와 Virtex-7 VX485T FPGA를 이용하여 각각 구현하고 비교 분석하였다. 처리할 수 있는 영상의 해상도는 수평, 수직 모두 1,024로 설정했고, 총 화소값 변화량(G)은 식 (4)의 절대값을 이용해 구하고 임계값 80을 기준으로 경계선 여부를 판단하였다. 그리고 영상의 입력과 소벨 연산 결과 영상의 출력을 위한 인터페이스 방식은 AXI4 스트림 프로토콜(AXI4-Stream Protocol)[17, 18]을 적용하였다. 기존 회로와 제안된 회로는 동일한 입력에 대해 동일한 결과를 출력하며 그림 8는 해상도가 1,024 × 1,024 인 영상에 대한 동작 사례를 보인다.

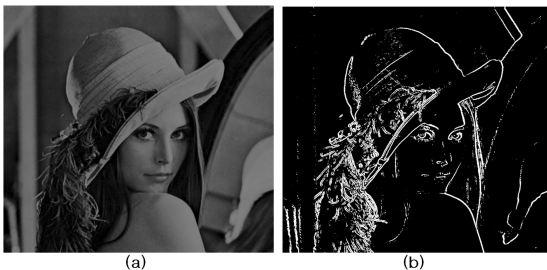


Fig. 8 Result of edge detection using proposed design (a) input image, (b) output image

표 1은 기존 방식과 제안된 방식으로 구현된 회로에 대한 FPGA 자원 사용률을 소요되는 SRAM, FF(Flip flop), LUT(Lookup table)별로 보여준다. BRAM_18K [19]는 Xilinx사의 7-시리즈 FPGA에서 제공되는 내장 SRAM 블록으로 1개당 18[Kb] 용량을 가지며 주로 FIFO 구현에 사용된다.

Table. 1 Resource utilization

		BRAM_18K (SRAM)	FF	LUT
Conventional	Used	1,026	28,188	102,771
	Utilization	49.80[%]	4.64[%]	33.85[%]
Proposed	Used	1,028	28,578	103,980
	Utilization	49.90[%]	4.70[%]	34.25[%]

FF:flip-flop, LUT: lookup table

제안된 회로를 구현하기 위해 기존 회로 대비 FPGA의 자원 사용이 약간 증가하였음을 알 수 있다. 1개의 소벨 연산 회로, 3개의 레지스터, 입력력 인터페이스 회로, 2개의 FIFO를 4개로 분리하기 위한 주변 회로 등이 추가되었지만, SRAM 블록은 단지 2개, FF와 LUT는 각각 1.4%와 1.2% 정도 증가하여 추가적인 자원 소요가 상당히 적다는 것을 알 수 있다.

그림 9는 동작 성능을 평가하기 위해 회로의 동작을 도식화하고 성능 지표를 설정한 것이다. 그림에서 t_{in} 는 입력 영상의 모든 화소들이 회로에 입력되는데 걸리는 시간을, t_{d1} 는 입력 영상의 첫 번째 화소값이 입력되는 시점과 출력 영상의 첫 번째 화소값이 출력되는 시점의 시간 차이, t_{d2} 는 입력 영상의 마지막 화소값이 입력되는 시점과 출력 영상의 마지막 화소값이 출력되는 시점의 시간 차이를 의미한다. t_{op} 는 $(t_{in} + t_{d1})$ 으로 영상 입력의 시작으로부터 결과 영상 출력의 종료까지 걸리는 시간을 의미하며, 회로의 윤곽선 검출 성능을 의미한다. 각 시간은 해당 동작을 수행하는데 소요된 클록의 수로 표현한다.

파이프라인이 적용되어 평균적으로 매 클럭마다 1회의 소벨 연산이 수행된다면, 해상도가 1,024 × 1,024 인 영상에 대한 기존 회로와 제안된 회로의 t_{op} 는 각각 1,048,576(=1,024×1,024)[clk]와 524,288(=1,048,576/2)[clk]를 약간 초과할 것이다.

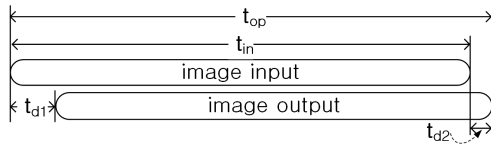


Fig. 9 Performance indices

Table. 2 Performance analysis

	Conventional(clk)	Proposed(clk)	Decrease(%)
t_{op}	1,053,706	529,419	50.2
t_{\in}	1,053,693	529,405	50.2
t_{d1}	2,073	1,048	50.6
t_{d2}	13	14	108

표 2는 기존 방식과 제안된 방식으로 구현된 회로의 t_{op} , t_{in} , t_{d1} 그리고 t_{d2} 을 소요된 클럭의 수로 보여준다. 기존 방식 대비 제안된 방식의 회로에서 t_{in} 과 t_{d1} 이 약 1/2 정도로 감소된 것을 볼 수 있다. t_{d2} 는 입력 영상의 마지막 화소값이 입력되고, 이 때 그림 4와 E의 레지스터들에 채워진 화소값에 대해 소벨 연산을 수행하여 출력 영상의 마지막 화소값을 출력하는데 소요되는 시간으로, 두 경우 모두 매우 작으며 제안된 회로가 1클럭 더 소요된다. 회로의 유평선 검출 성능을 의미하는 t_{op} 역시 제안된 방식의 회로가 기존 방식 대비 약 1/2 정도로 감소되어, 성능이 약 2배 정도 향상되었음을 알 수 있다. 그리고 기존 회로와 제안된 회로의 t_{op} 는 각각 앞서 언급한 1,048,576[clk]와 524,288[clk]를 조금만 초과했음을 알 수 있다.

IV. 결 론

본 논문에서는 소벨 연산을 이용한 영상의 유평선 검출을 고속으로 수행하기 위한 기존의 FPGA 기반 회로를 개선하여 약 2배의 성능을 갖는 회로 구조를 제안하였다. 기존의 방식은 유평선 검출 대상 영상의 화소값이 한 개씩 순차적으로 접근 가능한 상황에 적합하며, 제안된 방식은 대상 영상의 화소값이 동시에 두 개씩 접근 가능한 상황에 적합하다. 따라서 제안된 방식은 유평선 검출 대상 영상이 이미 메모리에 저장되어 있고 이를 하드웨어를 이용하여 처리하려는 상황에 적합하다.

기존의 회로와 제안된 회로를 모두 구현하여 기존의 회로 대비 제안된 회로가 필요로 하는 추가의 FPGA 자원이 많지 않음을 확인하였고, 각 회로의 성능을 한 개의 영상을 처리하는데 소요되는 클럭 수로 평가하여 제안된 회로가 기존의 회로 대비 약 2배 정도 빠르다는 것을 확인하였다.

REFERENCES

- [1] C. Zhang, N. Zhang, W. Yu, S. Hu, X. Wang, and H. Liang, "Improved Canny-based algorithm for image edge detection," in *Proceeding of 2021 36th Youth Academic Annual Conference of Chinese Association of Automation*, Nanchang, China, pp. 678-683, 2021.
- [2] L. Wang and Y. Sun, "Improved Canny edge detection algorithm," in *Proceeding of 2021 2nd International Conference on Computer Science and Management Technology*, Shanghai, China, pp. 414-417, 2021.
- [3] M. Weidong, T. Ying, Y. Yuxin, and Z. Yun, "Target Extraction Method Based on Improved Saliency Algorithm and Sobel Edge," in *Proceeding of 2020 5th International Conference on Mechanical, Control and Computer Engineering*, Harbin, China, pp. 1509-1512, 2020.
- [4] T. Wu, L. Wang, and J. Zhu, "Image Edge Detection Based on Sobel with Morphology," in *Proceeding of 2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference*, Xi'an, China, pp. 1216-1220, 2021.
- [5] S. Chen and X. Yang, "An Enhanced Adaptive Sobel Edge Detector Based on Improved Genetic Algorithm and Non-Maximum Suppression," in *Proceeding of 2021 China Automation Congress*, Beijing, China, pp. 8029-8034, 2021.
- [6] C. T. Johnston, K. T. Gribbon, and D. G. Bailey, "Implementing Image Processing Algorithms on FPGAs," in *Proceeding of 11th Electronics New Zealand Conference*, New Zealand, pp. 118-123, 2004.
- [7] G. -X. Yao, "Design of edge detection algorithm for image sobel based on FPGA," in *Proceeding of 2015 4th International Conference on Computer Science and Network Technology*, Harbin, China, pp. 851-853, 2015.
- [8] C. -S. Park and H. -S. Kim, "FPGA Implementation for Real Time Sobel Edge Detector Block Using 3-Line Buffers," *Journal of Institute of Korean Electrical and Electronics Engineers*, vol. 19, no. 1, pp. 10-17, Mar. 2015.
- [9] T. M. Khan, D. G. Bailey, M. A. U. Khan, and Y. Kong,

- “Real-time edge detection and range finding using FPGAs,” *International Journal for Light and Electron Optics*, vol. 126, no. 17, pp. 1545-1550, Sep. 2015.
- [10] N. M. Yusoff, I. S. A. Halim, N. E. Abdullah, and A. A. A. Rahim, “Real-time Hevea Leaves Diseases Identification using Sobel Edge Algorithm on FPGA,” in *Proceeding of 2018 9th IEEE Control and System Graduate Research Colloquium*, Shah Alam, Malaysia, pp. 168-171, 2018.
- [11] Z. Xiangxi, Z. Yonghui, Z. Shuaiyan, and Z. Jian, “FPGA implementation of edge detection for Sobel operator in eight directions,” in *Proceeding of 2018 IEEE Asia Pacific Conference on Circuits and Systems*, Chengdu, China, pp. 520-523, 2018.
- [12] K. Zhang, Y. Zhang, P. Wang, Y. Tian, and J. Yang, “An Improved Sobel Edge Algorithm and FPGA Implementation,” in *Proceeding of 8th International Congress of Information and Communication Technology*, Karachi, Pakistan, pp. 243-248, 2018.
- [13] N. Nausheen, A. Seal, P. Khanna, and S. Halder, “A FPGA based implementation of Sobel edge detection,” *Microprocessors and Microsystems*, vol. 56, pp. 84-91, Feb. 2018.
- [14] R. K. Megalingam, M. Karath, P. Prajitha, and G. Pocklassery, “Computational Analysis between Software and Hardware Implementation of Sobel Edge Detection Algorithm,” in *Proceeding of 2019 International Conference on Communication and Signal Processing*, Chennai, India, pp. 529-533, 2019.
- [15] J. -H. Lim and J. -Y. Ryu, “Edge Detection Control of Color Images Using FPGA,” *Journal of Institute of Control, Robotics and Systems*, vol. 25, no. 10, pp. 936-941, Sep. 2019.
- [16] H.L. Sneha, “The Why and How of Pipelining in FPGAs” [Internet]. Available: <https://www.allaboutcircuits.com/technical-articles/why-how-pipelining-in-fpga/>.
- [17] *AXI4-Stream Protocol Specification*, ARM, 2010.
- [18] *Vivado Design Suite AXI Reference Guide*, Xilinx, 2017.
- [19] *7 Series FPGAs Memory Resources*, Xilinx, 2019.



류상문(Sang-Moon Ryu)

전기및전자공학 박사
 군산대학교 IT정보제어공학부 교수
 ※ 관심분야 : FPGA 기반 임베디드 시스템, FPGA 기반 영상 처리